Research Paper | Computer Science | E-ISSN No : 2454-9916 | Volume : 3 | Issue : 2 | Feb 2017

TESTING OF VULNERABLE SOURCE CODE IN WEB APPLICATIONS

Priyanka Dnyaneshwar Patil

Computer & Science Department, North Maharashtra University, Jalgaon, Maharahtra, India.

ABSTRACT

The security of web application is a a main problem nowadays. This occurs due to code which are sometimes vulnerable, written in unsafe languages like PHP. Source code static analysis tools and Data mining tools are a solution to find vulnerabilities. There are some techniques generated to remove these vulnerabilities like static analysis tools and data mining. These techniques has successfully detected the vulnerabilities and also removed the vulnerabilities occurring in these languages. But the problem arises due to false positives i.e if any vulnerability has occurred but actually it is not the vulnerability in real fact e.g SQL Injection then in this study testing is performed to checked whether the detected vulnerability is really the vulnerability or it has occurred due to false positives in an application. This study also creates the report of this process.

KEYWORDS: Automatic protection, data mining, false positives, validation, software security, static analysis, web applications, software testing.

1. Introduction

Web application appear in many forms. Also PHP is open source language but problem arises in security of Web application. Many kind of problems arises due to this security problem. Sometime the system might predict the false positives but in real it is not the bug then system tries to remove these bugs and shows that it is an error. The solution to this problem is to perform testing internally and check whether the bug reported is really a bug or not. We use the word vulnerability for any false positive that have been found into the system. The code which is found vulnerable firstly an attributes are collected then the vulnerabilities are classified in classes according to an attributes. Then the code is checked whether it is vulnerable code or not. If the code is found vulnerable then it is fixed and inserted into the right place of code. The information which is true is given out as false positive information. Note that we use the word true positive for vulnerability which is a real vulnerability and false positive for the vulnerability which is a real vulnerability.

2. Project Flow:

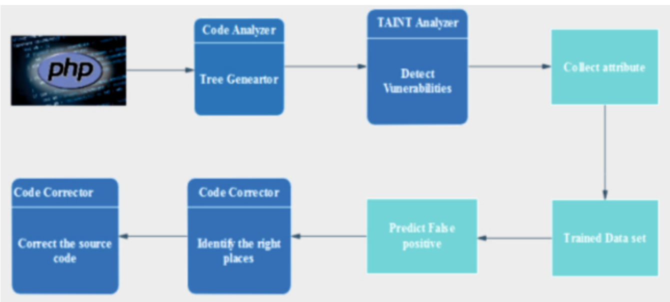

Fig. 1. Information flows that exploit web vulnerabilities

1. Firstly the PHP source code is given to tree generator into which it creates an AST i.e Abstract Syntax tree done by lexer and parser.

2. After an AST is created it is given to taint analyzer to create TST ie. Tainted symbol Tree and TEPT i.e Tainted Execution Path Tree .

3. Then, the system collects an attributes for the source code . An attributes can be concat, add, sum etc. These attributes are classified into classes like string manipulation, Validation and Query Manipulation.

4. There are 4 rule induction classifiers JRip, Prism, PART, Ridor used to find the correlation between vulnerable source code and attributes.

5. After then it is detected whether the code is really vulnerable or not using the classifiers like ID3,.J4/8, Random Tree, Random Forest, Logistic Regression, SVM, MLP, K-NN, Naïve Bayes, Bayes Net.

6. When the code is found to be vulnerable then an attributes are collected and grouped into classes and then the metrics are used to find best 3 classifiers for classification of vulnerabilities.

The vulnerabilities found can be of various types like SQLI(SQL Injection), Cross Site scripting(XSS), Remote File Inclusion(RFI), Local File Inclusion(LFI), Directory Traversal(DT), Path Traversal(PT), Source code Disclosure(SCD), PHP code Injection(PCI), OS command Injection(OSI).

3. Materials and Methods:

WAP (Web Application Protocol) is the protocol which is being used in this system. There are various kinds of tools that WAP make use. The WAP make use of static analysis and Data mining tools.Ther are various kinds of static analysis tools like Veracode, RIPS, Code sake Dwan, YASCA, VisualCodeGrepper, Devbug, Flawfinder, Brakeman, PMD, Xanitizer, taint analysis.

Veracode technology enables enterprises to test software providing greater security for organization, by scanning the binary code instead of source code. Veracode supports all widely used languages for desktop, wed and mobile application including like JAVA,.NET, C/C++ ETC.

RIPS is a static analysis tool to automatically detect vulnerabilities in PHP applications. By tokenizing and parsing all source files RIPS is able to transform PHP source code into a program model and to detect sensitive sinks that can be tainted by user inputs during program flow.

Brakeman is an open source vulnerability scanner specifically designed for Ruby on Rails application. Unlike web security scanners, Brakeman looks at the source code of web application. Brakeman scans an application code, it produces a report of all security issues it has found.

Code sake Dawn is an open source security source code analyzer designed for Sinatra, Padrino for Ruby on Rails applications.

PMD scans Java source code and tools for potential code problems.

Visual code grepper scans some software languages for security issues and for comments which may indicate defective code. The config file can be used to carry out additional checks for banned functions or functions which are commonly cause security issues.

Xanitizer scans Java for security vulnerabilities, mainly via taint analysis.

Taint analysis is used to identify variables that have been tainted with user controllable input and traces them to possible vulnerable functions also known as a 'sink'.If the tainted variable is given to sink without first being sanitized it is flagged as vulnerability. Some programming languages such as Perl and Ruby have taint checking built into them and enabled in certain situations such as accepting data via CGF(Control Flow Graph).

Data mining tools include RapidMiner, WEKA, Orange, KNIME, NLTK, Rattle etc.

RapidMimer is an open core model developed by the company that provides an integrated environment for machine learning, data mining, text mining, predictive analytics, business analytics. WEKA is collection of machine learning algorithm for data mining task. Weka features include machine learning, data mining, preprocessing, regression, clustering, associate rules etc.

Orange is an open source visualization and analysis tool. Data mining is done through visual programming or python scripting .

KNIME(Konstanz Information Miner) is open source data analytics, reporting platform. It integrates various components through its modular data pipelining concept.

NLTK is a leading platform for building Python programs to work with human languages data.

Rattle GUI is a free and open source software provides a graphical user interface for data mining using R statistical programming languages. Rattle allows dataset to be partitioned into training, validation, and testing.

Among all these tools static analysis uses taint analysis which is a tool of it in this project. Also data mining make use of AST which is tool of data mining. WAP does not use data mining to identify vulnerabilities but to predict whether the vulnerabilities found are really vulnerabilities or not.

There are some classifiers which are being used to categorize the vulnerabilities into classes. The classifiers are Logistic regression, Linear Regression, Decision trees, Perceptron, ID3,C4.5/J48,Random Tree, Random Forest, K-NN, Naïve Bayes, Bayes Net, MLP, SVM, etc.

Logistic regression is regression model where dependant variable is categorical. Cases where ouput occurs in 2 values like, "0"or "1","yes" or "no" etc.

Linear regression is approach for modeling the relationship between various kind of variables.

Decision tree is a decision support tool that uses a tree like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

Perceptron is an algorithm for supervised learning of classifiers whose output is in 2 forms.

ID3 is metadata container most often used in conjunction with MP3 audio file format.

Random tree is a tree that is formed by stochastic process. Types of this random trees include uniform spanning tree, Random minimal spanning tree, Random binary tree, random recursive tree etc.

Random forest are an ensemble learning method for classification, regression and other tasks, that operate by construction a multitude of decision trees at training time and outputting the class that is mode of classes.

K-NN is a non-parametric method used for classification and regression.

Naïve Bayes classifiers requires a number of parameters linear in the number of variables in learning problem.

Bayes Net is a probabilistic graphical model that represents a set of variables alomg with their conditional dependencies via a directed acyclic graph.

Among all these classifiers mentioned only 3 best classifiers are selected based on attribute dependency.

Some Induction Rule classifiers are again used to find the correlation between the vulnerabilities found and an attributes collected from them. These classifiers include PRISM, PART, JRip, Ridor etc.

PRISM takes an input a training set entered as a file of ordered set of attributes and classifies input from separate file at start of program and results are output as an individual rules.

PART produces a set of rules called decision list which are ordered set of rules. A new data is compared with each rule in list and item is assigned a category of first matching rule.

Ridor generates default rule first and then exceptions for the default rule with least error rate. Then it generates the "best" exceptions for each exception and iterates until pure.

JRip class implements a propositional rule learner, repeated Increamental Pruning to produce Error Reduction.

4. Conclusion

In this paper we present the techniques for securing web application vulnerabilities. The techniques make use of 2 tools static analysis and data mining which are most effective methods and provides results faster and correct manner.

REFERENCES

1. Symantec, "Internet threat report. 2012 trends, vol. 18," Apr. 2013.
2. W. Halfond, A. Orso, and P. Manolios, "WASP: protecting web applications using positive tainting" IEEE Trans.
3. Softw. Eng., vol. 34, no. 1, pp. 65–81, 2008.
4. T. Pietraszek and C. V. Berghe, "Defending against injection attacks through context-sensitive string evaluation," in Proc. 8th Int. Conf. Recent Advances in Intrusion Detection, 2005, pp. 124–145.
5. X. Wang, C. Pan, P. Liu, and S. Zhu, "SigFree: A signature-free buffer overflow attack blocker," in Proc. 15th USENIX Security Symp., Aug. 2006, pp. 225–240.
6. J. Antunes, N. F. Neves, M. Correia, P. Verissimo, and R. Neves, "Vulnerability removal with attack injection," IEEE Trans. Softw. Eng. vol. 36, no. 3, pp. 357–370, 2010.
7. R. Banabic and G. Candea, "Fast black-box testing of system recovery code," Proc.7th ACM European Conf. Computer Systems, 2012, pp. 281–294.
8. Huang, Yao-Wen et al, "Web application securit by fault injection and behavior monitoring," Proc. 12th Int. Conf. World Wide Web, 2003, pp. 148–159.
9. Huang, Yao-Wen et al, "Securing web application code by static analysis tools and runtime protection ," Proc. 13th Int. Conf. World Wide Web, 2004,.
10. N. Jovanovic, C. Kruegel, and E. Kirda, "Security using alias analysis for static removal of web application vulnerabilities," in Proc. 2006 Workshop on Programming Languages and Analysis for Security, Jun. 2006, pp.27–36.
11. W. Landi, "Undecidability of static analysis," ACM Letters on Programming Languages and Systems, vol. 1, no. 4, pp. 323–337, 1992.
12. N. L. de Poel, "Automated security review of PHP web applications with static code analysis and Data mining," M.S. thesis, State University of Groningen, May 2010.